

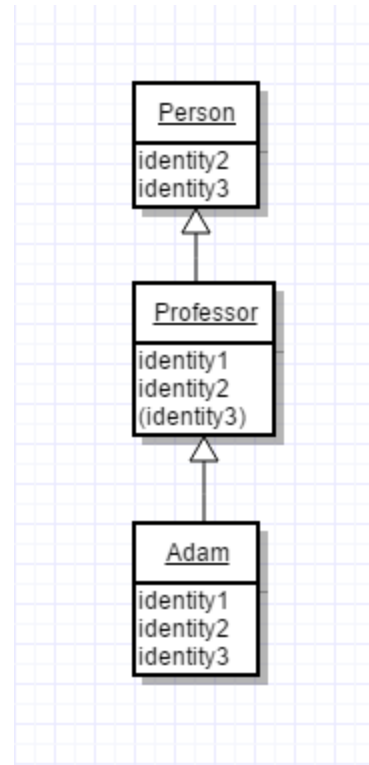
CSE 11 Winter 2016 Quiz 2

1. Given the following class definitions:

```
public class Person {
    public void identity2() {
        System.out.println("Person1");
    }
    public void identity3() {
        System.out.println("Person2");
    }
}

public class Professor extends Person {
    public void identity1() {
        System.out.println("Professor1");
    }
    public void identity2() {
        System.out.println("Professor2");
        identity1();
    }
}

public class Adam extends Professor {
    public void identity1() {
        System.out.println("Adam1");
    }
    public void identity2() {
        System.out.println("Adam2");
        super.identity2();
    }
    public void identity3() {
        System.out.println("Adam3");
    }
}
```



What is the output of the following code:

A.

```
Person jundt = new Adam();
jundt.identity1();
```

B.

```
Person jundt = new Adam();
jundt.identity2();
```

Output of A:

Output of B:

1. What is the Big-O complexity of the following code segments?

```
for(i = 0; i < 100; i++) {
    for(j = 0; j < 150; j++) {
        if(i == j) {
            System.out.println("match");
        }
    }
}
```

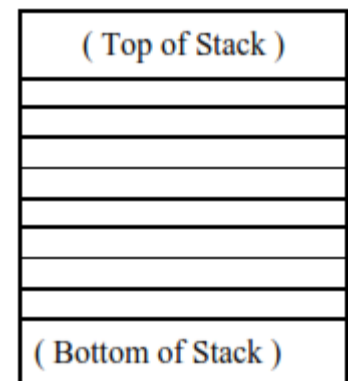
```
for(i = 0; i < n/2; i++) {
    for(j = 0; j < n; j++) {
        if(i == j) {
            System.out.println("match");
        }
    }
}
```

```
for(i = 0; i < n; i++) {
    for(j = 0; j <= i; j++) {
        if(i == j) {
            System.out.println("match");
        }
    }
}
```

2. Draw the order of methods on the stack up to the indicated point of program execution, *method2*. Note: the stack may not fill completely, also Stacks fill from the bottom to the top:

```
public class Example {
    private static int count = 10;

    public static void main (String[] args) {
        method1();
        Example ex1 = new Example();
        ex1.method2();
    }
    public static void method1() {
        Example ex2 = new Example();
        ex2.method4();
        ex2.method3();
    }
    public void method2() {
        // DRAW STACK THE FIRST TIME THIS IS REACHED
    }
    public void method3() {
        method2();
    }
    public void method4() {
        count--;
        return;
    }
}
```



3. What is the output from the following code:

```
public class Question4 {
    public static void main(String[] args) {
        Marker m1 = new Marker("red");
        Marker m2 = new Marker("black");
        m1.write("CSE");
        System.out.println(m2.getInk());
        m1.write("8B");
        m2.write("Midterm");
    }
}
```

Output

```
public class Marker {
    private static int ink = 2;
    private String color;
    public Marker() {
        this("black");
    }
    public Marker(String color) {
        this.color = color;
    }
    public static int getInk() {
        return ink;
    }
    public void write(String word) {
        if (ink!=0) {
            System.out.println(word+"/"+color+"/");
            ink--;
        }
        else
            System.out.println("Out of ink!");
    }
}
```

4. Draw the UML diagram for the following class:

```
public class Student {
    public String university = "UCSD";
    private String name;
    private int age;
    public Student() {
        this("Adam", "24");
    }
    public Student(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() {
        return name;
    }
    private void birthday() {
        age++;
    }
}
```

5. What is the output from the following program:

```
public class Food {  
    private String name;  
    private int price;  
  
    public Food(String name) {  
        this.name = name;  
        price = name.length();  
    }  
  
    public Food() {  
        this("");  
    }  
  
    public void print() {  
        System.out.println(name+": "+price);  
    }  
  
    public static void main(String[] args) {  
        Food rich = new Food("steak");  
        Food broke = new Food();  
        rich.print();  
        broke.print();  
    }  
}
```

Output

6. What is the output from the following code:

```
public class Grid {
    public static int energy = 0;
    public static void main(String[] args) {
        Grid grid = new Grid();
        NukaReactor r1 = new NukaReactor(1, grid);
        NukaReactor r2 = new NukaReactor(3, grid);
        NukaReactor r3 = r1;
        r3.upgrade();

        System.out.println(r1.getLevel());
        System.out.println(r2.getLevel());
        System.out.println(r3.getLevel());
        System.out.println(energy);
    }
}
```

Output

```
public class NukaReactor {
    private int level;
    private Grid grid;

    public NukaReactor(int level, Grid grid) {
        this.level = level;
        this.grid = grid;
        grid.energy += level;
    }

    public void upgrade() {
        grid.energy -= level;
        level++;
        grid.energy += level;
    }

    public int getLevel() {
        return level;
    }
}
```